Understanding the Mechanisms Modelled by Artificial Neural Networks for Hydrological Prediction

G. B. Kingston, H. R. Maier and M. F. Lambert

Centre for Applied Modelling in Water Engineering, School of Civil and Environmental Engineering,
University of Adelaide, Adelaide, SA, 5005, Australia. Email: gkingsto@civeng.adelaide.edu.au

Abstract: Artificial neural networks (ANNs) have been used increasingly in recent years for the prediction and forecasting of complex hydrological relationships. ANNs have been seen as an attractive alternative to process based modelling approaches, as they are able to extract an underlying relationship from the data when knowledge of the physical process is lacking. However, spurious correlations in the data can often lead to the incorrect underlying relationship being modelled and therefore care should be taken not to treat ANNs as black boxes, where data are input into the model and an output is generated with no knowledge gained on how the prediction was determined. It is essential that the mechanisms being modelled by the network are examined in order to provide some measure of confidence regarding the prediction. In this paper a number of investigations are carried out, using a synthetically generated data set, in order to assess the importance of extracting underlying relationships from trained ANNs. The results show that rule extraction techniques provide better information regarding the "correctness" of an ANN than conventional validation measures such as RMSE. Additionally, the methods proposed in this paper enable a comparison of the optimisation capabilities of different training algorithms. A comparison of a local search method (the backpropagation algorithm) and a global search method (the shuffled complex evolution algorithm) is presented.

Keywords: *Hydrology; Artificial neural network; Black box; Rule extraction; Garson's algorithm*

1. INTRODUCTION

Artificial neural networks (ANNs) have been used increasingly in recent years for the prediction and forecasting of complex hydrological relationships, including rainfall-runoff modelling, streamflow forecasting and prediction of water quality (ASCE, 2000; Maier and Dandy, 2000). As opposed to conventional modelling approaches, ANNs do not require an in depth knowledge of the driving processes, nor do they require the form of the model to be specified a priori. Therefore, they are often viewed as an appealing alternative when knowledge of the processes driving the hydrological phenomena is lacking. ANNs are able to extract a relationship between model inputs and outputs to provide quick and reasonably accurate predictions with a relatively minimum data requirement.

Despite these advantages, ANNs have frequently been criticised for operating as a "black box" (ASCE, 2000). An ANN is essentially a tool for the nonlinear mapping of inputs to outputs, where the primary purpose is to provide a prediction of system response rather than to gain an understanding of the causal interactions that generate the hydrological occurrence. A trained network does not provide any insight into how the prediction was determined and therefore cannot provide any measure of confidence regarding the prediction.

ANNs, like all data driven modelling approaches, are reliant on the quality and quantity of the data. If there are a large number of uncertainties associated with the data, as is often the case with measurements of environmental variables, or if the calibration data do not adequately represent the population of possible measurements, an ANN may not be capable of accurately estimating the underlying interactions that occur within the system. If some information regarding the modelled mechanics can be extracted from the network, an evaluation of the ANN can be carried out by comparing prior knowledge of actual physical processes with the relationships modelled by the network.

If ANNs are to become more widely accepted and reach their full potential as prediction models in hydrological modelling studies, some explanation capability is required. This paper presents the application of a rule extraction procedure in order to examine the advantages of rule extraction in ANN modelling.

return to menu

2. METHODS

The connection weights (free parameters) within an ANN provide the links between the inputs and outputs. Therefore, the relative contributions of the input variables in predicting the output are dependent on the magnitude and direction of the connection weights. An input will have a positive impact on the output if the input-hidden and hidden-output weights are of the same sign (i.e. both positive or both negative), whereas an input will have an inhibitory effect on the output if the signs of the input-hidden and hidden-output weights are opposite. Inputs can be identified as interacting with one another if the weights entering the same hidden node are of opposing signs.

Garson (1991) proposed a method for making use of the information contained in the network weights to determine the relative importance of each network input in predicting the output. However, this measure does not indicate the statistical significance of an input. Olden and Jackson (2002) proposed a randomisation method, incorporating Garson's algorithm, to statistically assess the contribution of inputs in the network. This method may be used to eliminate connections whose weights do not significantly influence the network output, thereby illuminating the significant interactions being modelled.

The methods used to gain information on network mechanics may be used to extract new rules regarding the physical processes of a system. However, the network needs to be globally optimised if the rules learnt are to reflect the true processes. Consequently the optimisation algorithm used also needs careful consideration.

2.1. Garson's Algorithm

Garson's measure of relative importance (Garson, 1991) uses the products of input-hidden and hidden-output connection weights to calculate the relative importance of each input variable. In the following explanation of Garson's algorithm the network weights are represented by $w_{i,j}$, where i is the node from which information is being passed and j is the node where the information is received.

Referring to Figure 1, the contribution of each input node to the output via each hidden node can be calculated as follows:

$$c_{1,A} = w_{1,A} \times w_{A,O} \quad \text{and} \qquad (1)$$
$$c_{1,B} = w_{1,B} \times w_{B,O} \quad \text{etc.}$$

From these values the relative contribution of each input node to the outgoing signal of each hidden node can be calculated by, for example:

$$r_{1,A} = \left|c_{1,A}\right| \Big/ \left(\left|c_{1,A}\right| + \left|c_{2,A}\right| + \left|c_{3,A}\right|\right) \qquad (2)$$

The relative importance of each input variable is then determined by:

$$RI_1 = S_1 \Big/ \left(S_1 + S_2 + S_3\right) \quad \text{where} \qquad (3)$$
$$S_1 = r_{1,A} + r_{1,B} \quad \text{and similar for } S_2 \text{ and } S_3$$

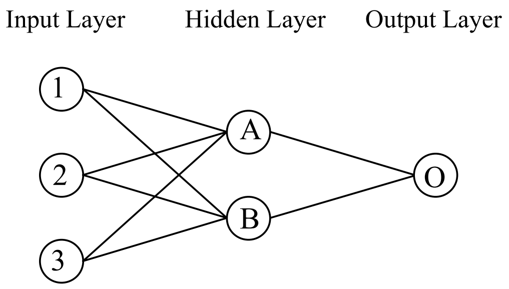

Figure 1. Example ANN structure

In this study the equations used in Garson's algorithm were modified such that the relative importance of hidden nodes could also be determined as follows:

$$rn_{1,A} = \left|c_{1,A}\right| \Big/ \left(\left|c_{1,A}\right| + \left|c_{1,B}\right|\right)$$
$$Sn_A = rn_{1,A} + rn_{2,A} + rn_{3,A} \qquad (4)$$
$$RIn_A = Sn_A \Big/ \left(Sn_A + Sn_B\right)$$

Because Garson's algorithm uses absolute values of input-hidden and hidden-output connection weights, it does not take into account the direction of the relationship between input and output variables. The use of absolute values can lead to misinterpretation of input importance. For example, input 1 may have a positive effect on the output through hidden node A, but an inhibitory effect on the output through hidden node B, and therefore the counteracting signs should act to diminish the overall importance of the input. However, Garson's algorithm only takes into account the absolute values of these effects, resulting in an increased rather than reduced overall importance.

2.2. Randomisation Approach

As Garson’s measure of relative importance can be misleading, the randomisation procedure introduced by Olden and Jackson (2002) uses an additional measure to assess the importance of network inputs, namely the overall connection weight as defined below. A variation of Olden and Jackson's (2002) randomisation procedure was carried out in this study, where the overall connection weights were used to assess the importance of inputs and the modified version of Garson’s algorithm was used to assess the relative importance of hidden nodes. The approach was carried out as follows:

1. construct an ANN using original input and output data with randomly generated initial weights;
2. train the network and calculate and record:
    - (a) input-hidden-output node connection weights: the product of input-hidden and hidden-output connection weights, e.g $c_{1,A}$ (see Garson’s algorithm);
    - (b) overall connection weight: the sum of the input-hidden and hidden-output connection weight products for each input variable, e.g. $C_1 = c_{1,A} + c_{1,B}$;
    - (c) relative importance of each hidden node as calculated by (4), e.g. RIn_A;
3. randomly permute the original target variable;
4. construct an ANN using the randomised targets;
5. repeat steps 2, 3 and 4 a large number of times.

In this approach the network targets are randomly shuffled to remove any structure between the inputs and targets. When the network is trained the resulting weight vector represents the case when there is no relationship between inputs and outputs and only spurious correlations are modelled. If the overall connection weight of an input, C_i, is greater than 95% of the randomised overall connection weights for the same input then the input can be considered to be significant with a 95% confidence level. The significance of a hidden node, j, may be determined in a similar way by comparing the relative importance of the node, RIn_j, to the 95th percentile randomised value. If inputs or hidden nodes are found to be insignificant they will subsequently be pruned from the network. This process can be repeated until only significant inputs and hidden nodes remain.

2.3. Network Training

Network “training” is the process of iteratively adjusting the connection weights such that a predetermined objective function is minimised and the best fit of the model predictions to the observed data is obtained. It can be compared to the calibration of a mathematical model.

The nonlinear characteristics of ANNs lead to the existence of multiple optima on the solution surface. However, there is currently no training algorithm that can guarantee the global optimal solution as opposed to converging on a local minima. Local or global optimisation algorithms may be used to train an ANN. Backpropagation, a first order local method, is currently the most widely used algorithm for optimising feedforward ANNs (Maier and Dandy, 2000). This algorithm is based on the method of steepest descent, where the network weights are updated according to:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \gamma_n \mathbf{d}_n \qquad (5)$$

where **w** is the vector of connection weights, γ is the step size and **d** is a vector defining the direction of descent. This algorithm is an effective way of optimising weights, however, like all local search methods, it is susceptible to becoming trapped in local minima in the error surface. Global methods have the ability to escape local minima as they employ random search techniques to allow the simultaneous search for an optimum solution in several directions. They are often more computationally intensive than local search techniques, but with improving computer technologies, the use of global optimisation methods is increasing. Duan et al. (1992) developed the shuffled complex evolution (SCE) algorithm that uses multiple “simplexes”, started from random locations in the search space, to direct the search towards the global optimum. At periodic stages of the search, the points in the simplexes are shuffled together to ensure that information is shared and that each simplex is not conducting an independent search of the global optimum.

Comparisons of the abilities of different training algorithms to find the global optima are generally based on the closeness of fit between predicted and observed outputs. However, there may be many combinations of network weights that will result in similar network performance. When applied to test cases with known relationships, rule extraction algorithms should aid in the comparison of optimisation techniques as they are able to determine whether the training algorithm used has adequately estimated the correct underlying function rather than merely generated good fitting results.

3. CASE STUDY

3.1. Data

Autoregressive (AR) models are commonly used to model hydrological time series data. The autoregressive model, AR(9), given by (6), was used to generate a set of synthetic time series data which were, in turn, used to demonstrate the importance of rule extraction. This model was selected as it depends on more than one input variable and has known dependence attributes.

$$x_t = 0.3x_{t-1} - 0.6x_{t-4} - 0.5x_{t-9} + \varepsilon_t \tag{6}$$

In the above equation ε_t is a normally distributed random noise component with mean of 0 and standard deviation of 1. The use of synthetic data enables the capabilities of the proposed method to be investigated without the complication of other sources of error, such as insufficient data, or the omission of significant inputs.

3.2. Investigations

An ANN was constructed to predict the above model. Although the model output only depends on inputs x_{t-1}, x_{t-4} and x_{t-9}, 15 inputs from x_{t-1} to x_{t-15} were included in the ANN. The network included 1 hidden layer with 5 hidden layer nodes as a starting point.

Assessment of Randomisation Procedure

The randomisation approach described in Section 2.2 was carried out in order to determine which inputs and hidden nodes made a significant contribution to the prediction of x_t and therefore, find the optimal network structure for modelling the AR(9) process.

The success of the randomisation approach relies upon the network's ability to converge on a (near) global minimum of the objective function rather than a local minima, as the connection weights of a network that has converged on a local minimum will differ from those that have globally converged. The backpropagation algorithm (Rumelhart et al., 1986) including momentum, and the shuffled complex evolution (SCE) method (Duan et al., 1992) were used for training the ANN so as to compare the success of the randomisation approach using each search method.

Comparison of Training Algorithms in Estimating the Underlying Relationship

The rule extraction algorithms were used to compare the relative predictive capabilities of the two training algorithms employed. The overall connection weights between input and output variables were examined to determine how accurate each training algorithm was in estimating the correct underlying relationship.

The ability to handle noisy data is one of the benefits of ANNs. The networks were trained on data generated by (6), which includes a random noise component, and the performance of each training algorithm was assessed in its ability to model the correct underlying relationship in the presence of noise. In order to do this a second data set was generated by (6), however this set did not include the random noise component. Performance measures given in the following sections are based on a comparison of the network output with this noise-free data set.

Predictive Performance

Generally, noisy observations must be relied upon to assess the predictive performance of an ANN. When data are noisy, performance measures such as the RMSE between predictions and observations may be misleading when trying to interpret how well the model fits the underlying function. If unnecessary inputs are included in the ANN, spurious correlations in the data may be modelled, resulting in an improved RMSE, however the incorrect underlying relationship will be inferred.

Two network structures, obtained during different stages of the randomisation procedure using the SCE algorithm, were compared in order to determine whether the use of an error measure such as the RMSE is sufficient to ascertain which network provides the best generalisation ability and therefore best predictive performance when presented with new data.

4. RESULTS & DISCUSSION

4.1. Determination of Significant Inputs and Hidden Nodes

The randomisation procedure for determining significant inputs and hidden layer nodes was iterative, where the ANN had to be retrained whenever inputs or hidden nodes were pruned from the network.

The results of the procedure are presented in Table 1, where the "optimal" network resulting from each training algorithm is presented in terms of the inputs determined to be significant, the number of hidden nodes and the RMSE between the predictions and the noise-free data set.

Table 1. Resulting "optimal" networks.

	Backpropagation	SCE
Significant Inputs	x_{t-1}, x_{t-4}, x_{t-9}	x_{t-1}, x_{t-4}, x_{t-9}
No. of hidden nodes	0	0
RMSE	0.125	0.100

The randomisation procedure was able to correctly identify which inputs were significant in predicting the output when each of the training algorithms were used. The model given by (6) is a linear process and therefore the most optimal network structure is that containing no hidden nodes. This was also correctly identified by the randomisation procedure.

It took 4 iterations to achieve the final network described in Table 1 using the SCE algorithm, whereas it took 7 iterations using the backpropagation algorithm. Therefore, it may be considered that the randomisation procedure is able to determine the optimal network structure more efficiently when used with the SCE algorithm. Additionally, the RMSE obtained by the backpropagation algorithm is slightly higher than that achieved with the SCE algorithm, although this may be a function of the stopping criterion used in each case.

4.2. Estimation of Underlying Relationship

Table 2 gives a comparison of the actual weights associated with each driving input and the overall connection weights in the "optimal" networks determined by the randomisation process using the alternative training algorithms. The error measure given in Table 2 was calculated by:

$$\sqrt{\frac{\sum_{i=1}^{3}(C_i - A_i)^2}{3}} \tag{6}$$

where C_i is the overall connection weight of input i and A_i is the actual weight for input i.

Table 2. Overall connection weights of "significant" inputs.

Input	Actual	Backpropagation	SCE
1	0.3	0.27	0.31
4	-0.6	-0.69	-0.68
9	-0.5	-0.57	-0.56
Error		0.065	0.057

It can be seen that each training algorithm was able to estimate the AR(9) model with good accuracy, especially considering that the training data contained added noise and a nonlinear activation function was used. The resulting network trained by the SCE algorithm was slightly more accurate than the network trained by backpropagation. However, as mentioned previously, this may be due to the stopping criteria used. The fact that the SCE algorithm was able to predict the underlying relationship with slightly more accuracy is also evidenced by comparing the output plots of the two networks as shown in Figure 2.

Predictive Performance

The results of two networks trained by the SCE algorithm were compared in terms of their overall connection weights and the RMSE between their outputs and the target data with added noise. The

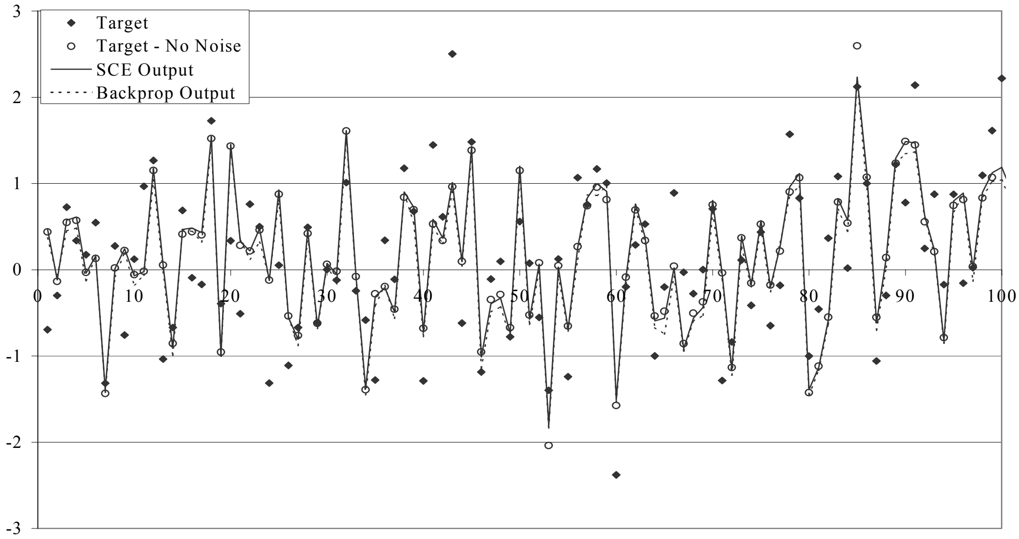

Figure 2. Plot of outputs from ANN with correct inputs and optimal structure, as determined by the backpropagation and SCE algorithms.

return to menu

first network has the optimal network structure as determined by the randomisation process (after 4 iterations). This network has 3 inputs and no hidden nodes. The second network is that obtained after only one iteration of the randomisation procedure and contains 15 inputs and 2 hidden nodes.

Table 3. Results of two different network structures.

	Overall Connection Weight		
Input	Actual	3 inputs, no hidden nodes	15 inputs, 2 hidden nodes
x_{t-1}	0.3	0.315	0.871
x_{t-2}	-	-	0.469
x_{t-3}	-	-	-0.319
x_{t-4}	-0.6	-0.680	-1.344
x_{t-5}	-	-	0.191
x_{t-6}	-	-	-0.246
x_{t-7}	-	-	-0.244
x_{t-8}	-	-	-0.388
x_{t-9}	-0.5	-0.555	-0.987
x_{t-10}	-	-	-0.322
x_{t-11}	-	-	0.318
x_{t-12}	-	-	3.498
x_{t-13}	-	-	3.530
x_{t-14}	-	-	-0.210
x_{t-15}	-	-	0.354
RMSE	-	0.544	0.542

The second network has incorrectly estimated the underlying relationship, as 12 unnecessary inputs were included in the model. Particular importance was given to inputs x_{t-12} and x_{t-13} when in fact they that were not required for the predictions. Also, the inclusion of 2 hidden nodes indicates that a nonlinear relationship has been estimated even though the actual model is linear. However, the RMSE of the second network is slightly less than that of the first network which may incorrectly suggest that the second network is an improvement over the first. By inspecting a plot of outputs from each of the networks it was indeterminable which structure was better able to model the actual AR(9) series as the outputs of each network were almost overlaying one another.

5. CONCLUSIONS

The selection of important inputs and appropriate network geometry significantly affect the predictive performance of an ANN, however neither is a trivial task. It was shown that rule extraction methods can be extremely beneficial in achieving an optimal model structure and that the randomisation procedure developed by Olden and Jackson (2002) is suitable for this process. The SCE algorithm displayed more efficiency in optimising the case study model. However, even in the presence of noise, it was demonstrated that both the backpropagation and SCE algorithms were able to represent the driving process with relative precision.

It was also demonstrated that the use of an error measure, such as the RMSE, is insufficient for assessing how well the underlying process is represented by the model when calibrated with noisy data. By using information obtained from the network weights a better assessment could be made as to how well the model had estimated the underlying function. With this capacity it is possible make predictions with greater confidence when new data are presented to ANNs.

6. REFERENCES

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. Artificial neural networks in hydrology. I: Preliminary concepts, *Journal of Hydrologic Engineering, ASCE*, 5(2), 115-123, 2000.

Duan, Q., S. Sorooshian, and V. K. Gupta. Effective and efficient global optimisation for conceptual rainfall-runoff models, *Water Resources Research*, 28(4), 1015-1031, 1992.

Garson, G. D. Interpreting neural-network connection weights, *Artificial Intelligence Expert*, 6, 47-51, 1991.

Maier, H. R., and G. C. Dandy. Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications, *Environmental Modelling and Software*, 15, 101-124, 2000.

Olden, J. D., and D. A. Jackson. Illuminating the "black box": a randomisation approach for understanding variable contributions in artificial neural networks, *Ecological Modelling*, 154, 135-150, 2002.

Rumelhart, D. E., J. L. McLelland, and PDP Research Group. *Parallel distributed processing, explorations in the micro structure of cognition*, Vol. 1: Foundations, MIT Press, Cambridge, Massachusetts, 1986.